Journal of Computational Mathematics, Vol.??, No.?, 200?, 1–8.

A global root-finding method for high dimensional problems[*]

Fabrizio Castellano
Dipartimento di Fisica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
Email: fabrizio.castellano@polito.it

Abstract

A method to solve the problem $f(x) = 0$ efficiently on any n-dimensional domain Ω under very broad hypoteses is proposed. The position of the root of f, assumed unique, is found by computing the center of mass of an Ω-shaped object having a singular mass density. It is shown that although the mass of the object is infinite, the position of its center of mass can be computed exactly and corresponds to the solution of the problem. The exact analytical result is implemented numerically by means of an adaptive Monte Carlo sampling technique which provides an exponential rate of convergence. The method can be extended to functions with multiple roots, providing an efficient automated root finding algorithm.

1. Introduction

Needless to say, the problem of finding the zeroes of a function is indeed one of the central issues in every branch of science, since many others can be reduced to it. Scalar equations of the form $f(\mathbf{x}) = g(\mathbf{x})$, where $\mathbf{x}$ is an n-dimensional vector, can be rewritten as $f(\mathbf{x}) - g(\mathbf{x}) = 0$. Nonlinear systems of equations $f_\alpha(\mathbf{x}) = 0$ can be rewritten as $F(\mathbf{x}) = 0$, where F is a vector valued function, and then reduced to a scalar equation by taking $|F(\mathbf{x})| = 0$.

When talking about finding the root of a function the dimension of the space plays a crucial role. For functions in one variable well estabilished and efficient algorithms exist, like the bisection, secant, Newton-Raphson methods and many others. These methods can either be higly efficient in terms of rate of convergence but may fail to converge in certain cases or can ensure convergence at the cost of a slower rate. The key feature in one dimension is that it is always possible to bracket the solution between two points, use such points as an estimate for the root and then improve above them. This is what allows to develop convergent algorithms. In two or more dimensions the problem is completely different because the boundary of any domain containing the solution is made up of an infinity of points rather than only two. Thus multidimensional root-finding problems are usually deterministically approached by methods involving local quantities, such as a guess of the solution and the derivatives of the function around it. These are extensions of the Newton-Raphson method[1] and may fail to converge if the function is not sufficiently well behaved in the domain of interest: rapidly oscillating or not-differentiable functions pose significant problems in the application of such methods.

As an alternative to deterministic methods, stochastic root finding algorithms provide an efficient way to cope with bad bahaved functions, and are the only available tool when the value of the function itself cannot be known exactly without a high computational cost[2]. Another promising alternative to root finding and other optimization problems is represented by genetic algorithms and evolutionary programming [3, 4], in which the "survival of the fittest" paradigm

[*] Received ???? ??, 200? / Revised version received ???? ??, 200? /

is used to select the solution among a population of initial candidates which are progressively mutated and discarded basing on a suitable fitness function.

One of the key problems affecting all the above-mentioned methods is that at least one initial guess of the solution is required, from which the method starts. Such initial guess is what ultimately determines if the method will converge to the correct solution or will get trapped inside a local minumum or other bad behaved regions. This weakness is due to the intrinsic locality of such methods, which try to improve on the available solution basing on informations regarding only a neighborhood of the current estimate.

In the present paper an exact analytical result that allows to directly compute the position of the root of a function is presented. Basing on this result a numerical method to find the solution is developed, which is intrinsically global and does not need an initial guess of the solution.

2. The center of mass property

Let us consider a scalar function f defined over an n-dimensional domain Ω having a unique zero at $\mathbf{x}_0 \in \Omega$. We will further assume that there exists a simply-connected open set $B^{\varepsilon}_{\mathbf{x}_0} \subset \Omega$,of arbitrarily small characteristic size ε, such that $\mathbf{x}_0 \in \partial B^{\varepsilon}_{\mathbf{x}_0}$ and over which f is differentiable. In practice we require f and Ω to be well behaved just over an arbitrarily small domain containing the root on its boundary. Given f and Ω satisfying the above conditions let us consider the family of functions

$$g_{k\eta}(\mathbf{x}) = \frac{1}{(f(\mathbf{x})^2 + \eta^2)^k} \tag{2.1}$$

where η is a real parameter and k is integer. Since by hypotesis $\mathbf{x}_0$ is the only root of f, these functions have an absolute maximum at that point, where they reach the value $\frac{1}{\eta^{2k}}$.

We can now switch to the problem of finding the position of the maximum of $g_{k\eta}(\mathbf{x})$, which is indeed equivalent to finding the zero of f. To solve this problem we borrow a very well known concept from classical mechanincs: the center of mass of a body with nonuniform mass density tends to lie near the points where the mass density is higher. With this in mind we can intuitively think that if a body has a mass density which diverges at a point, the center of mass should reasonably be near that point. Thus we regard the function $g_{k\eta}(\mathbf{x})$ as the mass density of an n-dimensional object, whose shape is the domain Ω, and compute the position of its center of mass when the parameter η becomes infinitely small.

We recall that the center of mass $\mathbf{x}_{CM}$ of an n-dimensional non-uniform body is computed as

$$\mathbf{x}_{CM} = \frac{\int_\Omega \mathbf{x}\rho(\mathbf{x})\mathrm{d}^n\mathbf{x}}{\int_\Omega \rho(\mathbf{x})\mathrm{d}^n\mathbf{x}} \tag{2.2}$$

where $\rho(\mathbf{x})$ is the mass density. In our case we take $\rho(\mathbf{x}) = g_{k\eta}(\mathbf{x})$ in the limit $\eta \to 0$, thus we write

$$\mathbf{x}_{CM} = \lim_{\eta\to 0} \frac{\int_\Omega \frac{\mathbf{x}\mathrm{d}^n\mathbf{x}}{(f(\mathbf{x})^2+\eta^2)^k}}{\int_\Omega \frac{\mathrm{d}^n\mathbf{x}}{(f(\mathbf{x})^2+\eta^2)^k}} \tag{2.3}$$

The key point is that the integrals in (2.3) can always be made divergent in the limit $\eta \to 0$ for sufficiently high values of k, depending on the dimension of Ω and the multiplicity of the root. Thus assuming a suitable value for k we can restrict the integration to any domain over which the integrals diverge without affecting the result, in partucular we choose to integrate over $B^{\varepsilon}_{\mathbf{x}_0}$

and then perform a variable change $\mathbf{x} - \mathbf{x}_0 \to \mathbf{x}$, obtaining

$$\mathbf{x}_{CM} = \mathbf{x}_0 + \lim_{\eta \to 0} \frac{\int_{B_0^\varepsilon} \frac{\mathbf{x}\mathrm{d}^n\mathbf{x}}{(f(\mathbf{x}+\mathbf{x}_0)^2+\eta^2)^k}}{\int_{B_0^\varepsilon} \frac{\mathrm{d}^n\mathbf{x}}{(f(\mathbf{x}+\mathbf{x}_0)^2+\eta^2)^k}} = \mathbf{x}_0 + \lim_{\eta \to 0} \Delta\mathbf{x}_\eta \tag{2.4}$$

Where B_0^ε is the ε-set having the point $\mathbf{x} = 0$ on its boundary. At this stage we can already see that the result of the computation is the searched solution $\mathbf{x}_0$ plus an η-dependent error term $\Delta\mathbf{x}_\eta$ which we shall show to go to zero as η vanishes.

Since B_0^ε is arbitrarily small we can subsitutue $f(\mathbf{x} + \mathbf{x}_0)$ with its lowest order Taylor expansion, which will depend on the multiplicity of the root. Without loss of generality we will assume that $f = |\mathbf{x} - \mathbf{x}_0|$ over $B_{\mathbf{x}_0}^\varepsilon$. In doing so we notice that, in the limit of vanishing η, at the denominator we are integrating a function with a singularity of the type $\frac{1}{|\mathbf{x}|^{2k}}$, while at the numerator the integrand has a $\frac{1}{|\mathbf{x}|^{2k-1}}$ singularity. The limit of the ratio of the two integrals is then zero and we can conclude that

$$\mathbf{x}_{CM} = \lim_{\eta \to 0} \frac{\int_\Omega \mathbf{x} g_{n\eta}(\mathbf{x})\mathrm{d}^n\mathbf{x}}{\int_\Omega g_{n\eta}(\mathbf{x})\mathrm{d}^n\mathbf{x}} = \mathbf{x}_0 \tag{2.5}$$

Thus the position of the root of f can be directly computed and corresponds to the center of mass of an Ω-shaped object having a singular mass density.

3. Numerical implementation

Equation (2.5) is exact and valid under broad assumptions and many numerical methods could be derived from it depending on how the integrals are actually handled. Any practical numerical implementation of (2.5) will be confronted with the fact that the integrals are divergent in the limit $\eta \to 0$, a feature which is a necessary condition for the result to hold. Even when using finite values of η we are left with the problem of computing $n + 1$ n-dimensional integrals, a task whose computational cost could make the proposed method not competitive with other approaches, in fact, any deterministic discretization of the integrals would imply a computational cost growing very rapidly with the number of dimensions of Ω. On the other hand statistical methods, such as Monte Carlo sampling, are much more effective for high dimensional problems because their rate of convergence is always $\frac{1}{\sqrt{N}}$ for N samplings, regardless of the dimension[5, 6]. Although a simple Monte Carlo sampling would then suffice, its rate of convergence would still be poor confronted with, e.g., the bisection method which has a rate of convergence of $\frac{1}{2^N}$ for N iterations, or Newton-Raphson methods which are even faster, provided that the initial guess is close enough to the solution to make them converge.

3.1. Parallel Monte Carlo sampling

In computing the integrals in (2.5) we can exploit a feature that will help speeding up things. We note that the two integrals to be computed are very similar, the only difference being that the function $g_{n\eta}$ is multiplied by $\mathbf{x}$ in the numerator. Moreover we know that the functions to be integrated reach very high values on a little region around the solution, while they are almost null everywhere else. Both features can be taken into account by developing an adeguate Monte Carlo sampling procedure in which the two integrals are computed in parallel: every time the numerator is sampled at the point $\mathbf{x}_i$, the denominator is sampled at the same point.

Thus after N samplings the estimated solution $\tilde{\mathbf{x}}_N$ is computed as

$$\tilde{\mathbf{x}}_N = \frac{\sum_{i=1}^{N} \frac{\mathbf{x}_i g_{k\eta}(\mathbf{x}_i)}{P(\mathbf{x}_i)}}{\sum_{i=1}^{N} \frac{g_{k\eta}(\mathbf{x}_i)}{P(\mathbf{x}_i)}} \tag{3.1}$$

where $\mathbf{x}_i$ is a sequence of points in Ω, generated according to the probability density P, which is used for the evaluation of both the numerator and the denominator. Thus if during the sampling the numerator has a sudden jump due to $\mathbf{x}_i$ lying near the singularity, the denominator will jump too, and this will reduce the fluctuations of the ratio, causing $\tilde{\mathbf{x}}_N$ to jump towards the exact value. This parallel-sampling choice has no formal justification besides the fact that it turns out to work well, in fact, even if we know that both integrals will converge to the respective solutions as $\frac{1}{\sqrt{N}}$, we cannot in principle tell anything about the rate of convergence of their ratio, because the two samplings are higly correlated in way that depends on the particular problem. Another useful property of (3.1) derives from the fact that we are computing the ratio of two integrals and thus the final result is independent from the normalization of $P(\mathbf{x})$, which is a major issue in Monte Carlo sampling techniques.

3.2. Uniform sampling

The simplest choice is to adopt a constant probability density $P(\mathbf{x}_i) = 1$ to generate the sampling points. In the following we shall show that in this case the method converges even in the case $\eta = 0$, and its rate of convergence is $\frac{1}{N^{\frac{1}{n}}}$. To do so we evaluate (2.4) using the parallel-sampling method with uniform probability density, after setting $\eta = 0$. We obtain

$$\tilde{\mathbf{x}}_N = \mathbf{x}_0 + \frac{\sum_{i=1}^{N} \frac{\mathbf{x}_i}{f(\mathbf{x}+\mathbf{x}_0)^{2k}}}{\sum_{i=1}^{N} \frac{1}{f(\mathbf{x}+\mathbf{x}_0)^{2k}}} = \mathbf{x}_0 + \Delta\mathbf{x}_N \tag{3.2}$$

where $\mathbf{x}_i$ is a sequence of uniformly distributed points inside B_0^{ε} and $\Delta\mathbf{x}_N$ is the error after N samplings.

If we now consider the quantity $|\Delta\mathbf{x}_N|$ we can write

$$|\Delta\mathbf{x}_N| \leq \frac{\sum_{i=1}^{N} \frac{1}{|\mathbf{x}_i|^{m-1}}}{\sum_{i=1}^{N} \frac{1}{|\mathbf{x}_i|^{m}}} \tag{3.3}$$

since $f(\mathbf{x}+\mathbf{x}_0)^{2k} \propto |\mathbf{x}|^m$, for some m, on an arbitraryly small domain around $\mathbf{x} = 0$.

We can rearrange the elements of the sequence $|\mathbf{x}_i|$ in order to $|\mathbf{x}_1|$ to be the smallest value and then write

$$\sum_{i=1}^{N} \frac{1}{|\mathbf{x}_i|^m} = \frac{1}{|\mathbf{x}_1|^m} + \sum_{i=2}^{N} \frac{1}{|\mathbf{x}_i|^m} \tag{3.4}$$

Now substituting (3.4) in (3.3) we obtain

$$|\Delta\mathbf{x}_N| \leq \frac{\frac{1}{|\mathbf{x}_1|^{m-1}} + \sum_{i=2}^{N} \frac{1}{|\mathbf{x}_i|^{m-1}}}{\frac{1}{|\mathbf{x}_1|^{m}} + \sum_{i=2}^{N} \frac{1}{|\mathbf{x}_i|^{m}}} \,. \tag{3.5}$$

Since the points $\mathbf{x}_i$ are uniformly distributed over the n-dimensional domain Ω the average distance between two points, in the limit $N \to \infty$, will be of the order of $\frac{1}{N^{\frac{1}{n}}}$ and thus $|\mathbf{x}_1|$ will be of that order too. Then, in the limit of large N, the term $1/|\mathbf{x}_1|^m$ in (3.4) will diverge while the rest of the summation will remain finite, we can then say that

$$\lim_{N\to\infty} |\Delta \mathbf{x}_N| \leq \lim_{N\to\infty} \frac{\frac{1}{N^{\frac{m-1}{n}}}}{\frac{1}{N^{\frac{m}{n}}}} = \lim_{N\to\infty} \frac{1}{N^{\frac{1}{n}}} = 0 \tag{3.6}$$

and thus the sequence $\tilde{\mathbf{x}}_N$ converges to $\mathbf{x}_0$ with a rate of convergence of $\frac{1}{N^{\frac{1}{n}}}$.

Thus in the case of uniform sampling the method is (almost surely) convergent but its rate of convergence is poor, depends on the dimension of the space and in particular becomes lower for higher dimensional problems.

3.3. Adaptive sampling

The efficiency of Monte Carlo sampling techniques is highly enhanced when the distribution of sampling points resembles the function to be integrated, thus we developed an adaptive-sampling method in which the probability density is varied during the process in order to maximize the efficiency. The key feature that allows such a strategy is again the fact that we are computing the ratio of two integrals and thus we are allowed to use non-normalized probability densities for the generation of sampling points. In particular, the parallel-sampling tecnique allows us to change the probability during the process: a peaked probability distribution is chosen for the generation of the sampling points, and its shape is gradually varied in order to concentrate the sampling around the solution.

The method proceeds as follows: a gaussian probability density of the form

$$P^i(x_1, ..., x_n) = \prod_{j=1}^{n} e^{\frac{(x_j - \tilde{x}_j^i)^2}{(\sigma_j^i)^2}} \tag{3.7}$$

is used for the distribution of sampling points, where x_j is the jth-component of $\mathbf{x}$, $\tilde{x}_j^i$ is the estimation of the solution at step i and σ_j^i is a suitable measure of the fluctuations of $\tilde{x}_j$ up to step i. When the process starts, σ_j^0 is set to a value such that the gaussian is flat over Ω. The estimation $\tilde{\mathbf{x}}^0$ loses all meaning in this case and can be set to an arbitrary point. After a few samplings, a new estimation of the solution is available, as well as an estimation of its fluctuations. These values are then used to update the probability density P^i. The key point is that as the sampling proceeds, the estimation comes closer to the exact solution and the fluctuations are reduced, thus P^i is gradually sharpened and centered around $\mathbf{x}_0$. This makes P^i resemble the functions to be integrated, making the sampling more efficient and further reducing the fluctuations, which in turn make P^i more peaked around the solution.

This adaptive-sampling technique must be tuned carefully. If the variance of the gaussian is reduced too fast, it can happen that the region around $\mathbf{x}_0$ is never sampled and thus the process converges to a random point and fails to find the correct solution. Conversely, if the variance is reduced too slowly, ther is no real advantage in using an adaptive algorithm and the results are the same as using a constant uniform probability density. In practice the key parameters are the number of samplings between two successive updates of P^i and the way σ is computed, a suitable choice has been found to be

$$\sigma_j^i = \frac{1}{N} \sum_{k=0}^{N-1} |\tilde{x}_j^{i-k} - \tilde{x}_j^{i-k-1}| \tag{3.8}$$

where the number of samplings N over wich the mean is performed determines the speed at which the variance of the gaussian is reduced, which in turn affects the rate of convergence.

Figure (3.1) shows three one-dimensional test problems (top figures) against which the method has been tested in order to check its convergence. In these examples P is updated every $N = 5$ samplings and $\eta = 10^{-8}$ The graphs at the bottom show typical convergence curves from which it can be seen that the rate is approximatively exponential (continuous line). It can also be noted that the parameter σ (dashed line) usually overestimates the actual error and can thus be taken as a conservative estimate. Actually, Figs. 3.1a and 3.1c show exceptions to this behaviour: this is caused by the finite value of η which introduces an intrinsic uncertainty in the determination of the solution and thus the error curves level at a certain value; anyway such error can be made arbitrarily small by reducing η. A similar test has been performed for multidimensional problems in Fig. 3.2 where the error $|\tilde{\mathbf{x}} - \mathbf{x}_0|$ is plotted against the number of Monte Carlo samplings. The function considered is the n-dimensional equivalent of Fig. 3.1b. It can be seen that the exponential rate of convergence is mantained in every dimension, differing only in the slope of the function. This is due to the fact that as the number of dimensions is increased, the parameter N in (3.8) has to be increased in order to ensure convergence. In this case $N = 10$ for the 1D problem and $N = 50$ for the 5D problem.

3.4. Automated root search

We have shown that the adaptive sampling method dramatically increases the rate of convergence with respect to the uniform sampling, at the cost of introducing the possibility that it could fail to converge in some cases. This weakness can actually be effectively turned into a benefit. The failure of the adaptive-sampling tecnique derives from having introduced some locality in the method, indeed, using a gaussian probability density corresponds to limiting the domain considered for the computation of integrals in (2.5). This locality can then drive the method to converge to local minima or to points inside regions where f reaches very low values. Actually the worst case would be that of a function f which has two roots in Ω: in this case the function $g_{k\eta}$ would have two singularities that would make the uniform-sampling method not converge to any of the two. On the other side, the adaptive-sampling method, due to its tendency to drive the probability density around points where f has very low values, will converge to one of the two solutions.

This feature can be exploited to develop an automated algorithm that sequentially discovers all the roots of f. Consider a function f defined on Ω which has a number of roots at points $\mathbf{x}_1, ..., \mathbf{x}_M$. The adaptive-sampling method then the method will converge to one of the solutions, say $\mathbf{x}_1$. A this point if the region surrounding $\mathbf{x}_1$ is removed from the domain Ω and the process is repeated, the method will converge to another solution, say $\mathbf{x}_2$. This loop can be iterated in order to find all the roots of f.

4. Summary and conclusions

We have proposed a global root-finding algorithm which allows to compute the position of the root of a scalar function over an n-dimensional domain under very broad assumptions. An exact analytical result is given which transforms the inverse problem of finding the root of a function to a direct computation of its position, interpreted as the center of mass of an n-dimensional Ω-shaped object with singular mass density. In view of this result a parallel

Monte Carlo sampling method has been developed and proven to converge with a rate of $\frac{1}{N^{\frac{1}{n}}}$. The method is then extended in order to employ adaptive sampling of the integrals, providing a numerical method with exponential rate of convergence which can be further extended to an automated root-search algoritm applicable to functions with multiple roots.

The proposed methods, being based on integral quantities, are global in nature and applicable to a wide variety of cases in which the function is bad behaved (not differentiable, singular, rapidly oscillating, ...) or the domain Ω is not regular (not simply connected, irregular shape, ...).

This paper is a demonstration of the possibilities of the proposed method, further work is in progress in order to improve its convergence rate and range of applicability. A wide variety of methods can be derived from equation (2.5), depending on the specific choice of the $g_{k\eta}$ function and the specific method used to compute the integrals, and this could open the way towards more efficient methods.

We are grateful to Fausto Rossi and David Taj for stimulating and fruitful discussion.

References

[1] J. M. Ortega, W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Society for Industrial Mathematics, 1987

[2] R. Pasupathy, B. W. Schmeiser, *Proceedings of the 2003 Winter Simulation Conference*, 2003

[3] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin, 1992

[4] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989

[5] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, Methuen, London, 1964

[6] M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods Vol. 1: basics*, John Wiley and Sons, 1986

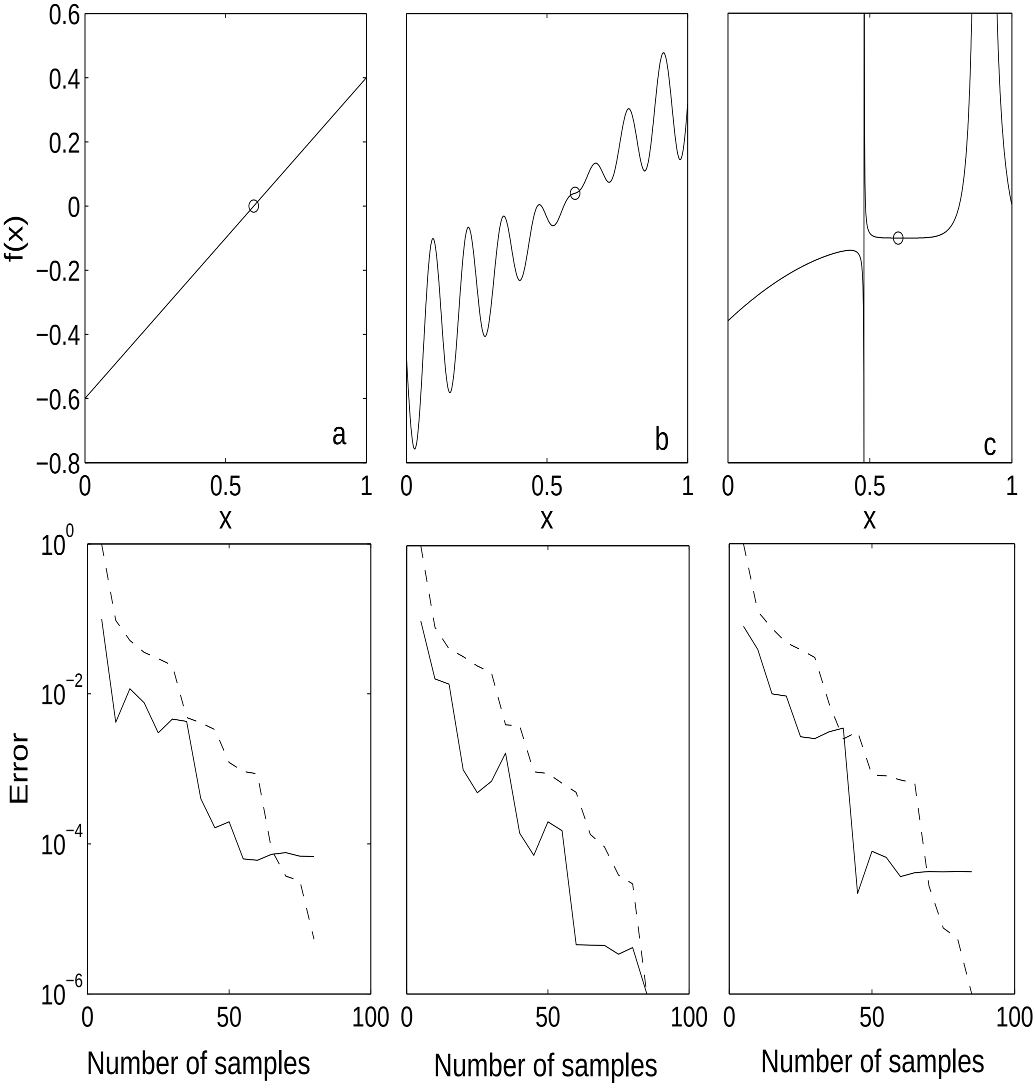

Fig. 3.1. Three one dimensional root-finding test problems solved by the adaptive Monte Carlo sampling method. Figures on the top show the test functions, each of which has a zero at $x_0 = 0.6$. Figures on the bottom row show the corresponding convergence tests for the adaptive Monte Carlo sampling method: dashed lines show the the fluctuation σ of the solution, the continuous line is the actual error $|\tilde{x} - x_0|$.

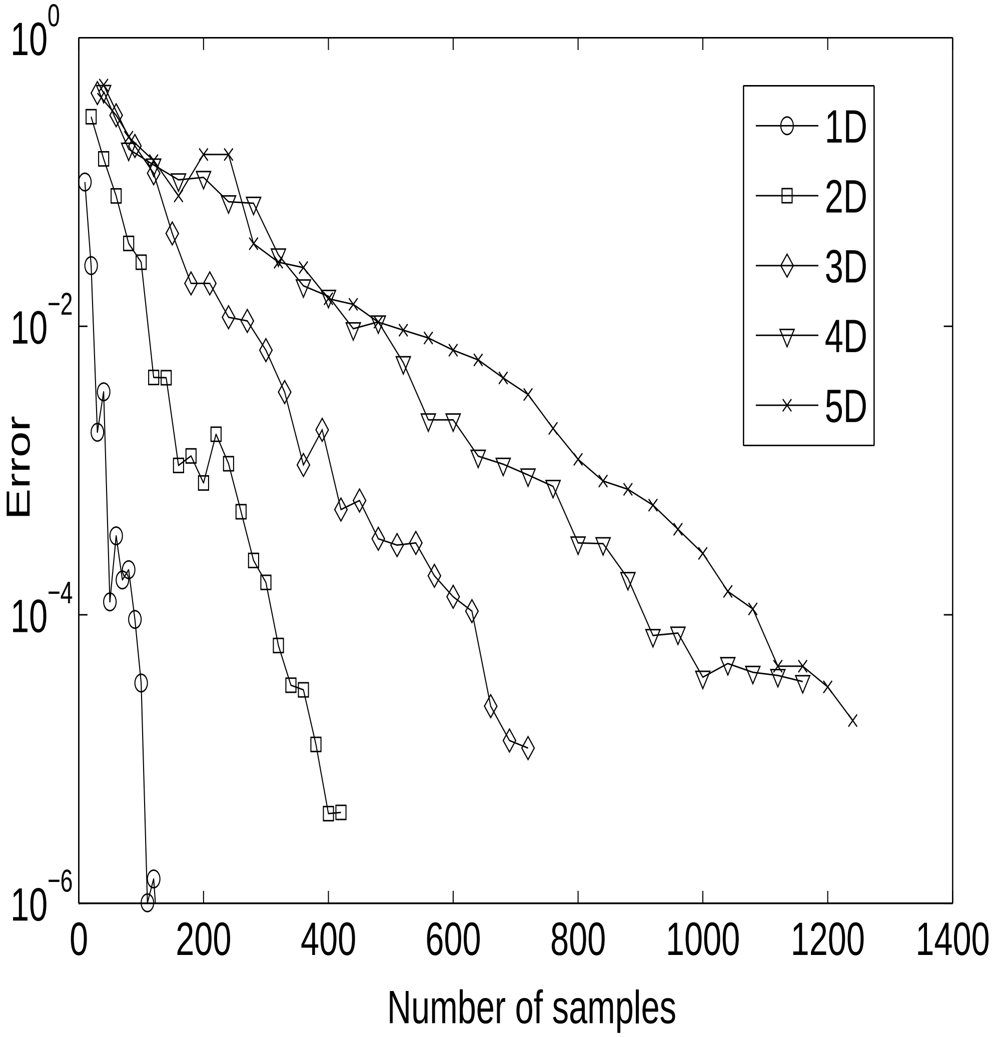

Fig. 3.2. Convergence test for multidimensional problems. The rate of convergence is lower for high dimensional cases, but remains exponential.